



# Cost Implications for a Discrete Mathematics Intelligent Tutoring System

Ofentse Rice, George Anderson and Tallman Nkgau

Department of Computer Science

University of Botswana, Gaborone, Botswana

ophuti@gmail.com, georgeganderson@gmail.com, nkgautz@gmail.com

## Abstract

An Intelligent Tutoring System (ITS) is software that dynamically modifies the presentation of educational material to students based on Artificial Intelligence. ITSs enable tutoring to be done by computers; teaching is customized for the student using the system, enabling faster learning; and there is a potential to enable scaling up, teaching of millions of students, using online platforms or platform independent programs which is not possible in the case of human tutoring. In this paper we focus on the cost implications of an ITS designed for teaching Discrete Mathematics at the undergraduate level at the University Of Botswana. The ITS discussed, DS-ITS, makes use of probabilistic graphical networks to relate core concepts of Discrete Mathematics concepts with verifiable outcomes (demonstrable knowledge). A student's understanding of a concept is inferred; it can be viewed as the cause of failing to answer a question correctly. Our contribution is in estimating the cost required to build a prototype, which can be used for certain Discrete Mathematics topics as taught at the undergraduate level. We compare the cost implications with the alternative of not having such a tool and instead relying purely on lecturers and teaching assistants. Our findings show the benefit of developing such a tool.

## 1 Introduction

Tutorials are used in schools as a form of content delivery, the University of Botswana being no exception. One-on-one tutoring has been found to be the most effective form of tutoring and for over many years has been provided by humans (Merrill, Reiser, Ranney, & Trafton, 1992). Intelligent Tutoring Systems (ITSs) have been developed so as to mimic human tutors through adaptation and personalization of tutoring content. The ITSs are meant to improve the demerits that are associated with human tutoring such as the cost that increases with the number of students to tutor, limited tutoring times and lack of scalability amongst others. In light of these advantages we are developing an ITS for Discrete Mathematics, offered by the Department of Computer Science in the University of Botswana,

Discrete Structures Intelligent Tutoring System (DS-ITS). This will augment human tutoring for this subject because it has many students and thus available tutors for it is taken by a large number of undergraduate students (220 students in the first semester of academic 2017/2018). This paper focuses on the cost implication of introducing such a tool to use in the department. We compared the development and deployment of DS-ITS to the current form of tutoring that is conducted by lecturers with the aid of teaching assistants.

The rest of this paper is organized as follows: Section 2 gives a background on Intelligent Tutoring Systems; Section 3 discusses related work; Section 4 discusses the cost of Intelligent Tutoring systems in general; Section 5 has a cost analysis, with results; Section 6 has discussion of the cost; Section 7 has a conclusion.

## 2 Intelligent Tutoring Systems

Intelligent Tutoring Systems simulate the way humans conduct one on one tutoring. ITSs collect the student learning needs and adapt tutoring content based on these learning needs. These learning needs or characteristics forms the student module/student profile which is the driving force of adaptation in an ITS. ITSs have been successfully deployed and are continuously being proposed because they are effective as a method of tutoring. ITSs have a generic architecture, which is made of four main interacting modules. The expert module, the student module, the tutor module and the user interface (Nkambou, Mizoguchi, & Bourdeau, 2010; Chen, MdYunus, Ali, & Bakar, 2008; Meshref, 2011; Nwana, 1990; Ralegaonkar, Thakare, & Thorat, 2015; Nouh, Nadarajan, & others, 2006; Chakraborty & Sinha, 2016)

An intelligent Tutoring System(ITS) is created to simulate human tutoring by identifying students learning needs and adapting tutorial content as per the student's needs (Lane, 2006). Several ITSs have been developed and deployed for research and for use in different educational domains such as economics, language learning and programming (Jones, 1985).

The generic architectural structure of an ITS has four modules (Chen, MdYunus, Ali, & Bakar, 2008; Meshref, 2011; Nwana, 1990; Ralegaonkar, Thakare, & Thorat, 2015).

- i. The Student Module which is the driving force of adaption in an ITS. It stores the learning characteristics of a student as per the goal of an ITS and uses these to adapt the tutorial content for each student. This module helps the ITS to understand the unique learning abilities of a student and continuously updates them as the student uses the system.
- ii. The Domain Module; an explicit representation of the domain knowledge. All learning materials for the domain are stored in this module.
- iii. The Tutor Module makes all pedagogical decisions and stores all knowledge about how to teach. It dynamically adapts the teaching instructions during the tutorial session using the student and knowledge module.
- iv. The User Interface, which is the point of interaction for the student and the system.

Development and deployment of an ITS comes with costs, but the main question is, do the costs of having such a tool beat the traditional tutoring method (lecturers and teaching assistants)? In the next section, we look at work that tries to answer this question for various environments.

### 3 Related Work

Although ITSs can be very beneficial, their development and deployment comes at a cost. Different expertise are required to develop the different ITS modules mentioned about. To develop an effective ITS knowledge in cognitive science, rule-based programming and high level of computer science skills are required hence a larger cost of deployment (Gilbert, Blessing, & Guo, 2015). According to literature; to develop an hour instruction for an ITS can take up to 200-1000 hours (Gilbert, Blessing, & Guo, 2015; Devasani, Gilbert, & Blessing, 2012). Contributing to the cost of development is the cost of deployment. To deploy these systems, computer and or mobile devices are required. These devices must have specific computing power to be able to run AI based ITSs. The cost of development has however reduced as technology costs have and continue to decline. Research has been carried out to find ways of reducing development cost by reducing cost of instruction development and reduction in skills required to develop different modules. This has been achieved by researchers developing different tools that assist in the development of the ITSs resulting in decreased production time, low-level or non-programming skills and novices in cognitive science (Cognitive Tutor Authoring Tools, n.d.).

The ASSISTment tool builder is one of the authoring tools that are aimed at reducing cost of production of ITS. The tool builder has been designed such that it provides web based tools for teachers to create content for their students. Content creation can be done with ease and at a relative short time using simplified GUIs (Heffernan, et al., 2006). The ASSISTment tool builder was tested by allowing non-programmers to create content and contents creation was relatively low compared to the 200 hours for an hour's instruction. The subjects managed to create content at an average time of 60 minutes for 2 minutes instructions.

Cognitive Tutor Authoring Tools is another group of authoring tool that aims at increasing authoring efficiency using an example based approach. CTAT have been widely used, approximately 220 research groups use these tools to develop their own ITSs and this acceptance signifies their ease of use in developing ITSs. We can infer from this that the cost of producing an ITS using CTAT is reduced (Devasani, Gilbert, & Blessing, 2012).

Another authoring tools that helps to create computer based instructions is the xPST (Extensible Problem Specific Tutor). No programming experience is required for one to use this tutor to build student interfaces rather third party software is used (Gilbert, Blessing, & Guo, 2015).

With the authoring tool the development time, cost of development and expertise required in the educational domains that they are developed for is greatly reduced. This suggest that it becomes cheaper and faster, thus it is cost effective to develop usable ITSs for tutoring in schools.

## 4 Cost of Delivering Intelligent Tutoring Systems

Literature suggests that developing that an ITSs is costly. The determining factors of the cost of developing an ITS are the skills and the time required to develop an ITS. This is so because to develop an effective ITS several skills sets at a high level are required. Developing each individual model requires people with different expertise; domain experts, pedagogy experts and skilled and experienced people and to obtain people with such skills comes at a cost. The development of an ITS is a difficult and time consuming tasks, research suggests that it takes approximately 200-1000 hours to create 1 hour worth of instruction. In the following subsection we discuss two ways of developing an ITS.

### 4.1 Authoring Tools

Different tools are used to ease and reduce the cost of development of ITSs Amongst the widely used authoring tools are the CTAT and the Extensible Problem Specific Tutor (xPST).

CTATs supports the development of two Cognitive Tutors and Example-Tracing Tutors (Alevan, McLaren, Sewall, & Koedinger, 2006). To build ITSs using CTATs authors do not need any programming skills because the CTAT uses an example based approach to develop the ITS. The author models the examples that will be used, demonstrating the correct and incorrect problem solving behavior. These examples are stored by the CTAT. Depending on the kind of ITS to be built the author either generalizes or annotate these examples. These examples are then used the bases of Example-Tracing Tutors and or guide the development of Cognitive Tutors (Alevan, McLaren, Sewall, & Koedinger, 2006). CTAT provides the author with tools to speed up development such as the GUI Builder to drag and drop widgets and build a user interface, the behavior recorder that uses recorded examples to implement the example tracing function and an Eclipse editor to write the production rules. All these tools together with no-programming requirement simplifies the development of ITSs using CTAT (Alevan, McLaren, Sewall, & Koedinger, 2006).

xPST's aim is to quicken the creation of computer-base instruction while making sure that there are no programming skills required. With xPST authors can use third party software as a student interface. As the author manipulate the user interface xPST authoring tool creates hints and thus helps to quickly create a domain module (Gilbert, Blessing, & Guo, 2015).

The ASSISTment project is an ongoing project that aims at reducing cost of development by insuring that ITSs can be developed by everyone as easily as possible. It has an ASSISTment builder which is allows teachers with no programming skills to create required tutoring content for their students in the shortest time possible (Heffernan, et al., 2006). It achieves this by providing the authors with a form based GUI that allows the user to specify the problem interface by dragging and dropping high level widgets provided (Heffernan, et al., 2006).

## 4.2 Fully Customized

Our definition of a fully customized ITS is developing all the modules of an ITS in-house. In our case, we will define the full customization based on our DS-ITS prototype design. The DS-ITS uses a probabilistic graphical representation; Bayesian Networks (BN) for intelligence; Knowledge assessment and decision-making.

For the construction of the BN we use the libpgm 1.3 (Welcome to libpgm, n.d.), a Python Library that implements BN and the BN functionalities that we require for our student model. The creation of the student model involved looking for an appropriate open source BN library, learning the library and customizing it as required. All this requires at least intermediary programming experience and the understanding of the BN algorithm.

Creation of the domain model requires expert knowledge; the authors of the paper with the assistance of Discrete Mathematics lecturers are used as experts and pedagogy designers. Two lecturers and one teaching assistant knowledgeable about Discrete Mathematics were consulted. The user interface (input/output model) used requires intermediary programming skills making it possible for the researchers to develop. We decided to choose the fully customized option because the authoring tools had a considerable amount of functionality, which is not required, making them complex to work with, and more difficult to maintain. With the fully customized option, the researchers are in full control of critical aspects of the projects, such as the student model, as well as the overall design, which is important for growth beyond the prototype.

## 5 Cost Analysis

To analyze the cost benefits of having DS-ITS and purely relying on lectures and teaching assistants as (hereafter referred to traditional tutoring method) we look at the cost involved in using each method

and comparing these costs. We did a cost analysis of using these systems for one academic year so as to determine which is more cost effective.

## 5.1 Traditional Tutoring Method

The department of Computer Science in the University of Botswana uses teaching assistants and lab demonstrators to conduct tutorials for student. Because our interest lies in determining the cost implications of using DS-ITS, we will discuss the cost of traditional tutoring based on how Discrete Mathematics tutorials are conducted. Discrete Mathematics is one of the large courses in the department and can have approximately 220 students per semester. For tutorial groups the students are divided into 6 tutorial groups, with roughly 36 students per group. The groups are then assigned a lab demonstrator and a teaching assistant. This means that the ratio of tutor to student is 1 tutor to 18 students. We use a lab demonstrator of salary of BWP 160,000.00 per year as a sufficiently accurate approximation an average salary. Their duties include conducting tutorials, marking the tutorial assignments, be available to assist students outside scheduled tutorial times and do administrative work such as photocopying and invigilating tests/examinations. Teaching assistants are paid per hour to conduct the tutorials, and they earn BWP 85.00 per hour (an approximation of an average pay rate). The tutorial content is created by lecturers and they are expected to create sample solutions for the questions.

## 5.2 Fully Customized

The lab demonstrators earn approximately BWP 160,000.00 and to determine the cost of their tutoring services for discrete mathematics we first determined the salary per hour. Eight (8) hours a day is a fair choice of hours because the demonstrators are expected to be in tutorials for 26 hours a week and the rest of the hours in a week a reserved for administrative tasks. These costing are calculated as shown in Table 1.

**Table 1. Lab Demonstrator earnings per hour**

Money Earned (BWP)	
Per Annum (12 months)	160,000.00
Per Month	13,333.33
Per Hour (8 hours in a day)	83.33

**Table 2. Cost of discrete mathematics tutoring for an academic year per demonstrator**

Cost for Tutoring Discrete Mathematics (BWP)	
Per Week (4 hours per week)	333.32
Per Annum (26 weeks)	8,666.32

An academic year is equivalent to 10 months, or 26 weeks; calculations are shown in Table 1. Lab demonstrators are paid a fixed salary at the end of each month and are expected to work on weekends; hence, we approximated a month to 4 weeks, or 20 days.

We then determined the cost of hours spent in the discrete mathematics tutorials Table 2, note that this time does not include the administrative task time for discrete mathematics because they are not well defined and they vary from week to week. Therefore, the total cost for demonstrators tutoring Discrete Mathematics for an academic year is

$$\begin{aligned}
 \text{Cost} &= \text{Number of demonstrators} \times \text{annum rate} \\
 &= 3 \times 8,666.32 \\
 &= 25,998.96
 \end{aligned}$$

Which evaluates to BWP 25,998.96 using the annum rate in Table 2. Teaching assistants earn BWP 85.00 per hour and work for 4 hours in Discrete Mathematics, thus the cost of tutoring discrete mathematics per teaching assistant is as shown in Table 3. Teaching assistants are paid an hourly rate and they work 4 hours discrete mathematics tutorials a week, a month has 4 weeks and we approximate a semester to 13 weeks.

**Table 3. Teaching assistant earnings for discrete mathematics tutoring**

Discrete Mathematics cost of Tutoring Teaching Assistants (BWP)	
Per Week (4 hours * 85.00)	340.00
Per Annum (26 weeks)	8,840.00

$$\begin{aligned}
 \text{Cost} &= \text{Number of Teaching Assistants} \times \text{annum rate} \\
 &= 3 \times 8,840.00 \\
 &= 26,520.00
 \end{aligned}$$

The total cost of this method of tutoring is approximately BWP 52,518.96 per annum as calculated in Table 4.

**Table 4. Cost of discrete mathematics tutoring for one academic year**

Discrete Mathematics cost of tutoring (BWP)	
Lab demonstrators	25,998.96
Teaching assistants	26,520.00
Total for one academic year	52,518.96

### 5.3 DS-ITS Approach

We chose to develop DS-ITS from scratch thus it falls under the fully customized form of ITS development. There are different cost models have been proposed for software development cost estimation (Keim, Bhardwaj, Saroop, & Tandon, 2014). Each model chosen to use is based on the size and complexity of the project thus we chose to use COCOMO (Constructive Cost Models) II for it is suitable for our project and is the most widely used model of estimation (Keim, Bhardwaj, Saroop, & Tandon, 2014; Heemstra, 1992). COCOMO II has different variants: Basic COCOMO for rough estimates, Intermediate COCOMO for detailed estimate for small and medium sized projects and Detailed COCOMO for detailed estimate for large projects. The three variants of COCOMO II models are then characterized into three different modes of development based on the complexity of the project (Lane, 2006). The three different modes of development or software products are organic, semi-detached and embedded listed in increasing order of complexity. For each mode of development COCOMO II there are defined model parameter constants: a, b, c, and d which are influencing factors of development for each category of software product as given by Boehm in 1981 as cited in (Keshta, 2017).

To estimate the cost of development for DS-ITS prototype we used COCOMO II Basic for embedded systems. Basic COCOMO II was chosen for we want to give a rough estimation of the cost of developing DS-ITS. To calculate the cost of software development using this model one has to take into account:

- i. The effort (E); labor required to complete different tasks of development and is measured in person-months. a is a productivity coefficient and b is a scale factor coefficient (Keshta, 2017).

$$E = a(KLOC)^b$$

Where a=3.6, b=1.20, and KLOC is kilo-lines-of-code.

- ii. Schedule or time from development (TDEV); the amount of time needed to complete project measured in unit times.

$$TDEV = c(E)^d$$

Where c, d are constants based on model calibration for, thus for embedded software product c=2.5 and d=0.32.

We estimated our KLOC to be 2.6 as shown in Table 5. Using this value effort evaluates to:

$$E = 3.6(3.2)^{1.20} = 14.54 \text{ person months}$$

Then we calculate the duration required for project completion as:

$$TDEV = 2.5(14.54)^{0.32} = 5.9 \text{ months}$$

To estimate the cost of the project in Botswana Pula, we used the government payment scale for a software developer. A software developer is in Scale C2 and earns BWP 147,252.00 per annum. This means that a software developer earns approximately BWP 12,271.00 per month.

$$\begin{aligned} \text{Total cost of developmen} &= \text{months} \times \text{cost/month} \\ &= 5.9 \times 12,271.00 \\ &= 72,398.90 \end{aligned}$$

Thus it will cost the university approximately BWP 72,398.90 to implement an ITS.

**Table 5. Calculating KLOC for DS-ITS**

Activity	Lines of Code
Developing a Student Module	1000
Developing a Knowledge Module	1000
Developing a User Interface	1000
Integration of Modules	200

## 6 Discussion

The cost analysis approach we used to determine the implications of having DS-ITS in place did not include the cost of content creation. This is because in both tutoring methods we require experts to

create content for tutoring in this case our experts will be course lecturers. At a glance one might assume that the cost benefits of using the traditional method of tutoring outweighs the cost of introducing the DS-ITS tutoring method. This however it is not the case. The total cost for the traditional method is BWP 52,518.96per annum and that of developing a DS-ITS is BWP 72,398.90 once off payment. It is important to note that this estimation for the traditional method is for one academic year, this amount increases as the university continues to offer the course. Suppose the university offers discrete mathematics for the next five years, the amount of the traditional method accumulates with each year, while that if the DS-ITS remains constant (see Figure 1).

**Table 6. Cost of tutoring over a period of 5 years**

Number of years	Traditional Tutoring Method	DS-ITS
1	52,518.96	72,398.90
2	105,037.92	72,398.90
3	157,556.88	72,398.90
4	210,075.84	72,398.90
5	262,594.80	72,398.90

It also increases as the government increases salaries but the DS-ITS costing is a once off costing. Since content creation and update is left for the lecturers who are already in the university payroll system. For the traditional method, there is no scalability, the more the number of students the more the tutors, the more the tutors the more the cost of tutoring. Furthermore, with this current method the tutoring times are fixed, students cannot get extra help from the tutors outside the tutoring hours. This is because teaching assistants do not work outside their allocated tutoring times and it is difficult for the students to get extra help from the demonstrators as their free times may conflict with the times the students are free.

With the DS-ITS method, there is scalability. The students will be able to use this outside allocated tutoring times, at their own pace and even at their homes given they have computers.

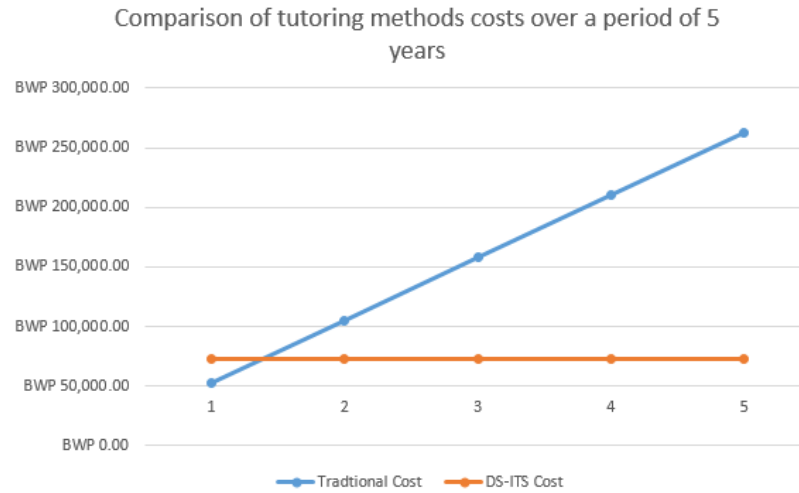
Furthermore, the cost of using an ITS is influenced by the development cost, and once developed it continues to be used at the same cost of development. The cost benefit of using a DS-ITS is clear; see Table 7.

**Table 7. Cost of using different methods for tutoring of an academic year**

DS-ITS	Traditional Tutoring Method
72,398.90	52,518.96

It is important to note that our study is meant to give a baseline for a cost-benefit of using human tutors versus using an ITS. Certain factors were not incorporated, such as potential software engineering costs involving project managers, requirements analysis, etc., as well as maintenance. Since this is a research project, the development costs would be absorbed as part of research activity, which involves lecturers as well as students. In addition, costs of human tutoring is actually much higher because of hours spent marking and administering the course, as well as consulting. There could also be a hybrid approach were human tutoring is augmented with an ITS. This scenario would not decrease demand for human tutors, and would enable students to improve their learning.





**Figure 1. Comparison cost for the two tutoring methods over a period of 5 years using data in Table 6.**

## 7 Conclusion

The cost analysis we carried out indicates that there is a cost benefit in using a tool like DS-ITS for tutoring. Both the university and the students benefit from such a tool. The university would be able to provide personalized tutoring students at an affordable cost and the students will greatly benefit from the flexibility and of tutoring that the tools brings them. Adopting such a system to augment the traditional tutoring system will be beneficial to department of Computer Science since the department has financial constraints to hire more human tutors. It is important to note that these are rough estimates, however in a case where the university wants to augment the ITS with the traditional method then the cost of production for the ITS remains lower and it still brings added advantages such as scalability with it.

## References

- Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2006). The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. *International Conference on Intelligent Tutoring Systems*, (pp. 61-70).
- Chakraborty, B., & Sinha, M. (2016). Student Evaluation Model Using Bayesian Network in an Intelligent E-Learning System. *Journal of Institute of Integrative Omics and Applied Biotechnology (IIOAB)*, 7, 2.
- Chen, T., MdYunus, A., Ali, W. Z., & Bakar, A. (2008). Utilization of Intelligent Tutoring System (ITS) in mathematics learning. *International Journal of Education and Development using ICT*, 4.
- Cognitive Tutor Authoring Tools*. (n.d.). Retrieved from <http://ctat.pact.cs.cmu.edu/>

- Devasani, S., Gilbert, S. B., & Blessing, S. B. (2012). Evaluation of two intelligent tutoring system authoring tool paradigms: Graphical user interface-based and text-based. *Proceedings of the 21st Conference on Behavior Representation in Modeling and Simulation*, (pp. 54-61).
- Gilbert, S. B., Blessing, S. B., & Guo, E. (2015, Sep). Authoring Effective Embedded Tutors: An Overview of the Extensible Problem Specific Tutor (xPST) System. *International Journal of Artificial Intelligence in Education*, 25, 428-454. doi:10.1007/s40593-015-0045-0
- Heemstra, F. J. (1992). Software cost estimation. *Information and software technology*, 34, 627-639.
- Heffernan, N. T., Turner, T. E., Lourenco, A. L., Macasek, M. A., Nuzzo-Jones, G., & Koedinger, K. R. (2006). The ASSISTment Builder: Towards an Analysis of Cost Effectiveness of ITS Creation. *FLAIRS Conference*, (pp. 515-520).
- Jones, M. (1985). Applications of artificial intelligence within education. *Computers & mathematics with applications*, 11, 517-526.
- Keim, Y., Bhardwaj, M., Saroop, S., & Tandon, A. (2014). Software cost estimation models and techniques: A survey. *International Journal of Engineering Research and Technology*, 3, 1763-1768.
- Keshta, I. M. (2017). Software Cost Estimation Approaches: A Survey. *Journal of Software Engineering and Applications*, 10, 824.
- Lane, H. C. (2006). Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning.
- Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2, 277-305.
- Meshref, H. (2011). An Intelligent Tutoring System for Logic Circuit Design Problem Solving. *International Journal of Computer Applications (0975-8887) Volume*.
- Nkambou, R., Mizoguchi, R., & Bourdeau, J. (2010). *Advances in Intelligent Tutoring Systems* (1st ed.). Springer Publishing Company, Incorporated.
- Nouh, Y., Nadarajan, R., & others. (2006). Intelligent tutoring system-bayesian student model. *2006 1st International Conference on Digital Information Management*, (pp. 257-262).
- Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4, 251-277.
- Ralegaonkar, A. A., Thakare, A. P., & Thorat, S. B. (2015). An Intelligent Tutoring System using Bayesian Network-A Review. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCE)*, 4, 143.
- Welcome to libpgm. (n.d.). Retrieved from <https://pythonhosted.org/libpgm>